

**NOTA:** esta breve guía no pretende y no debe reemplazar el uso del HELP

### **OPERACIONES ARITMÉTICAS**

Las operaciones aritméticas son: suma, resta, multiplicación, división, potenciación, división entera.

+, -, \*, /, ^, %/% or %%

### **OPERADORES DE COMPARACIÓN**

OPERADOR	SIGNIFICADO
>	mayor que
<	menor que
>=	mayor o igual
<=	menor o igual
==	igual que
!=	distinto que

### **OPERADORES LÓGICOS**

Los operadores lógicos son: "y", "o", "no" y "o exclusivo".

e1 & e2  
e1 | e2  
! e1  
xor(e1, e2)

### **ALGUNAS FUNCIONES**

exp(x)  
log(x, base=exp(1))  
log10(x)  
sqrt(x)                      raíz cuadrada  
abs(x)                      valor absoluto  
ceiling(x)                   menor entero >= x.  
floor(x)                    mayor entero <= x.  
trunc(x)                    entero más cercano a x entre x y 0, inclusive,  
e.g., trunc(1.5) es 1, y trunc(-1.5) es -1. trunc es como floor para  
valores positivos y como ceiling para valores negativos  
round(x,n)                   redondea con n dígitos

### **OPERADOR DE ASIGNACIÓN**

Asigna un valor a un nombre.

<-  
expresión <- valor

También son operadores de asignación \_ y =

expresión \_valor

expresión = valor

*Pero se recomienda el uso de <- como operador de asignación.*

### **PARA GENERAR UN VECTOR**

Para obtener el vector  $(x_1, \dots, x_n)$  se usa `c(x1,...,xn)` que concatena los elementos  $x_1, \dots, x_n$  generando un vector.

Ej: `c(1,2,5)` genera el vector  $(1,2,5)$

En general `c(x1,...,xn)` concatena los elementos  $x_1, \dots, x_n$  en una lista si los  $x_i$  no son todos del mismo tipo.

### **FUNCIÓN SAMPLE**

`sample(x,size, replace=FALSE,prob)`

Toma una muestra de tamaño "size" de elementos de "x" con o sin repetición según se indique TRUE o FALSE.

"prob" es un vector adicional que indica la probabilidad de obtener `c/u` de los elementos que figuran en "x", si no se pone nada se asume que todos tienen la misma probabilidad.

Ejemplos

`sample(1:k,n)` genera n números al azar entre 1 y k (si  $n \leq k$ )

`sample(1:k,n,T)` genera n números al azar con repetición entre 1 y k (si  $n > k$ )

`sample(c(10,20,30,40,50),3)` elige al azar una permutación de 3 números entre 10,20,30,40 y 50 sin repetición.

Si no se indica "size" ni "replace" asume que "size" es la longitud del vector x y por default asume `replace = FALSE`

`sample(6)` genera una permutación sin repetición de los 6 números 1,2 3, 4, 5 y 6

### **FUNCIÓN SUM**

`sum()` devuelve el resultado de la suma de los valores presentes en el argumento

Ejemplos

`sum(1:5)` devuelve 15 que es el resultado de sumar los naturales de 1 a 5

`sum(c(2,3,5,7))` devuelve 17 que es el resultado de la suma de los elementos que figuran en el argumento

`sum(c(2,3,5,7)==c(2,4,5,6))` devuelve 2 que es el número de elementos coincidentes entre los dos vectores

### **USO DE FOR**

`for(i in 1:n)` recorre los números naturales desde 1 hasta n

Ejemplo: con el siguiente algoritmo se obtiene el resultado de sumar los números naturales del 1 al 5

```
suma<-0
for(i in 1:5)
{
  suma<-suma+i
}
suma
```

### **USO DE IF**

`if(condición)` expresión

ejemplo: `if(x>0) y<-1`, le asigna a y el valor 1 si x es mayor que 0

`if(cond) expresión else expresión alternativa`

ejemplo: `if(x>0) y<-1 else y<-0`, le asigna a la variable `y` 1 si `x` es mayor que 0 y el valor 0 en caso contrario

### PARA OBTENER SECUENCIAS

**seq:** Crea un vector de números equiespaciados. El principio, el fin, el espacio entre dos números consecutivos o la cantidad de números de la secuencia pueden ser especificados

Generación de secuencias

- 1) `from:to`
- 2) `seq(from, to)`
- 3) `seq(from, to, by=)`
- 4) `seq(from, to, length=)`
- 5) `seq(along)`

Ejemplos:

```
1.1)
> 1:5
[1] 1 2 3 4 5
1.2)
> 5:1
[1] 5 4 3 2 1
1.3)
> 1.1:5
[1] 1.1 2.1 3.1 4.1
2.1)
> seq(5)
[1] 1 2 3 4 5
2.2)
> seq(-5)
[1] 1 0 -1 -2 -3 -4 -5
3.1)
> seq(0, 1, 0.1)
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
3.2)
> seq(1,20,2)
[1] 1 3 5 7 9 11 13 15 17 19
4.1)
> seq(-pi, pi, length=5)
[1] -3.141593 -1.570796 0.000000 1.570796 3.141593
```

**rep:** repite un vector `x` una cantidad determinada de veces (`times`) o hasta lograr la longitud especificada (`length.out`).  
`rep(x, times, length.out)`

Ejemplos

```
rep(0,10)
[1] 0 0 0 0 0 0 0 0 0 0
> rep(1:4,2)
[1] 1 2 3 4 1 2 3 4
```

Si `times` es un vector de la misma longitud de `x`, indica el número de repeticiones para cada componente de `x`.

```
> rep(1:4,c(2,2,2,2))
[1] 1 1 2 2 3 3 4 4
> rep(1:4, length.out=18)
```

```
[1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2
```

### **PARA HACER GRÁFICOS**

**plot(x,y)** Si x e y son vectores de la misma longitud representa los pares ordenados con puntos

### **PARA ADICIONAR LÍNEAS, PUNTOS O SEGMENTOS EN UN GRÁFICO**

lines(x, y)

points(x, y)

segments(x1, y1, x2, y2) adiciona un segmento a un gráfico uniendo el punto (x1,y1) con el punto (x2,y2), si x1,y1,x2,y2 son vectores de longitud k adiciona k segmentos

### **ARGUMENTOS OPCIONALES EN PLOT**

plot(x,y,type="")

type= "p", "l", "b", "o", "s" , "h" y "n", produce puntos, líneas, ambos, ambos superpuestos, escalones , barras verticales o nada

plot(x,y,xlim=,ylim=)

xlim e ylim son vectores que dan los límites para el recorrido de x e y en el gráfico

Ej: plot(x,y,xlim=c(1,5),ylim=c(0,0.3))

### **PARA AGREGAR UN TÍTULO**

title(main = "", sub = "", xlab = "", ylab = "", ...)

main (en la parte superior del gráfico)

sub (en la parte inferior del gráfico)

xlab (en el eje x)

ylab (en el eje y)

### **PARA AGREGAR TEXTO EN LOS MÁRGENES**

mtext(text="", side=3, line=0)

side (puede ser 1,2,3 o 4 según si el texto va en la parte inferior, izquierda, superior o derecha)

line=0 lo pone del lado de afuera pegado al margen

line=k con k>0 lo pone afuera del margen a distancia k

line=k con k<0 lo pone por adentro del margen a distancia k

## **DISTRIBUCIONES**

### **Instrucciones para la binomial**

Si  $X \sim \text{Bi}(n,p)$

para calcular  $P(X=k)$

**dbinom(k, size=n, prob=p)**

Ej: Si  $x \sim \text{Bi}(5,0.1)$  para calcular  $P(X=3)$

dbinom(3,5,0.1)

```
[1] 0.0081
```

para calcular  $P(X \leq k)$

```
pbinom(k, n, p) calcula la probabilidad acumulada  
P( $x \leq 3$ )  
pbinom(3, 5, 0.1)  
[1] 0.99954
```

#### Instrucciones para la distribución Poisson

```
Si  $X \sim P(\lambda)$   
dpois(x, lambda= $\lambda$ ) calcula  $P(X=x)$   
ppois(x, lambda= $\lambda$ ) calcula  $P(X \leq x)$ 
```

#### Instrucciones para la distribución Normal

```
Si  $X \sim N(\mu, \sigma^2)$   
dnorm(x,  $\mu$ ,  $\sigma$ ) calcula  $f(x)$  (función de densidad)  
pnorm(q, mean= $\mu$ , sd= $\sigma$ ) calcula  $P(X \leq q)$   
qnorm(p, mean= $\mu$ , sd= $\sigma$ ) calcula  $x$  /  $P(X \leq x) = p$   
rnorm(n, mean= $\mu$ , sd= $\sigma$ ) genera una muestra al azar de tamaño  $n$  de una  
distribución  $N(\mu, \sigma^2)$ .  
El default es en todos los casos  $\mu=0$ ,  $\sigma=1$ 
```

#### Instrucciones para la distribución de Student

```
Si  $X \sim t_n$  ( $n$  grados de libertad)  
dt(x, n) calcula  $f(x)$  (función de densidad)  
pt(q, n) calcula  $P(X \leq q)$   
qt(p, n) calcula  $x$  /  $P(X \leq x) = p$   
rt(m, n) genera una muestra al azar de tamaño  $m$  de una distribución  
Student con  $n$  grados de libertad
```

#### Instrucciones para la distribución de Chi-cuadrado

```
Si  $X \sim \chi^2_n$  ( $n$  grados de libertad)  
dchisq(x, n) calcula  $f(x)$  (función de densidad)  
pchisq(q, n) calcula  $P(X \leq q)$   
qchisq(p, n) calcula  $x$  /  $P(X \leq x) = p$   
rchisq(m, n) genera una muestra al azar de tamaño  $m$  de una distribución  
Chi-cuadrado con  $n$  grados de libertad
```